

LISTING OF CLAIMS:

1. (Currently Amended) A method, implemented in a data processing system, for determining a complexity of an enterprise information resource management system, the enterprise information resource management system being used to contain an ontology into which a plurality of enterprise data assets are mapped, the ontology including a plurality of model constructs, the enterprise data assets including a plurality of assets constructs, and the mappings between the enterprise data assets and the ontology including a plurality of mapping constructs, the method comprising:

receiving (i) a quantity of distinct asset constructs, denoted by C_{ASSET} , (ii) a quantity of distinct mapping constructs, denoted by $C_{MAPPING}$, and (iii) a quantity of distinct model constructs, denoted by C_{MODEL} ;

evaluating a metric of complexity, denoted by M , for ~~[[an]]~~ the enterprise information resource management system having a capacity corresponding to C_{ASSET} , $C_{MAPPING}$, and C_{MODEL} , wherein the metric of complexity is evaluated according to a formula

$$M = f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X),$$

where f is a real-valued function of three or more real-valued parameters and X denotes ~~optional~~ zero or more additional parameters; and

using the metric M within a transaction processing system, for license of the enterprise information resource management system.

2. (Currently Amended) The method of claim 1 wherein the plurality of enterprise data assets include conformed assets that conform to a general data schema that uses element group asset constructs and element asset constructs.

3. (Currently Amended) The method of claim 2 wherein the general data schema ~~[[is]]~~ comprises a relational database schema, ~~[[and]]~~ the element group asset constructs ~~[[are]]~~ comprise database tables, and the element asset constructs ~~[[are]]~~ comprise columns of database tables.

4. (Currently Amended) The method of claim 2 wherein the general data schema [[is]] comprises an XML schema, [[and]] the element group asset constructs [[are]] comprise XML complex types, and the element asset constructs [[are]] comprise XML elements.
5. (Currently Amended) The method of claim 2 wherein the general data schema [[is]] comprises a Cobol Copy Book, [[and]] the element group asset constructs [[are]] comprise Cobol group items, and the element asset constructs [[are]] comprise Cobol elementary items.
6. (Currently Amended) The method of claim 2 wherein the ontology [[is]] comprises an ontology model, [[and]] wherein the model constructs include ontology classes and [[their]] properties of the ontology classes, and wherein the model constructs further include business rules that inter-relate the properties.
7. (Currently Amended) The method of claim 6 wherein the mapping constructs include mappings of element group asset constructs into ontology classes and further include mappings mapping of element asset constructs into ontology properties.
8. (Original) The method of claim 6 wherein the function $f(C_{ASSET}, C_{MAPPING}, C_{MODEL})$ is a step function of the form

$$f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X) = f_n(X), \text{ if } C_{n-1} < C \leq C_n,$$

where C is the total number of constructs, $C = C_{ASSET} + C_{MAPPING} + C_{MODEL}$, and where C_0, C_1, C_2, \dots are cutoff points.

9. (Original) The method of claim 6 wherein the function $f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X)$ is a step function of the form

$$f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X) = f_n(X), \text{ if } C_{n-1} < C \leq C_n,$$

where C is a weighted average $C = W_{ASSET} C_{ASSET} + W_{MAPPING} C_{MAPPING} + W_{MODEL} C_{MODEL}$ and where W_{ASSET} , $W_{MAPPING}$ and W_{MODEL} are respective weighting factors, and where C_0, C_1, C_2, \dots are cutoff points.

10. (Currently Amended) The method of claim 1 wherein the enterprise information resource management system generates results for tasks, and wherein the metric M also depends on [[the]] a number of distinct results generated and saved.

11. (Original) The method of claim 10 wherein the results include data transformations.

12. (Original) The method of claim 10 wherein the results include SQL queries.

13. (Original) The method of claim 10 wherein the results include XSLT scripts.

14. (Currently Amended) The method of claim 1 wherein the enterprise information resource management system generates scripts for producing reports, and wherein the metric M also depends on [[the]] a number of distinct report scripts generated and saved.

15. (Currently Amended) The method of claim 1 wherein the enterprise information resource management system records metadata, and wherein the metric M also depends on [[the]] a number of distinct metadata records.

16. (Currently Amended) A method, implemented in a data processing system, for determining a complexity of a metadata repository including a plurality of metadata constructs, the comprising:

receiving a quantity of distinct metadata constructs, denoted by C ;

evaluating a metric of complexity, denoted by M , for a metadata repository having a capacity corresponding to C , wherein the metric of complexity is evaluated according to a formula

$$M = f(C, X),$$

where f is a real-valued function of one or more real-valued parameters and X denotes ~~optional~~ zero or more additional parameters; and

using the metric M within a transaction processing system, for license of the metadata repository.

17. (Currently Amended) The method of claim 16 wherein the metadata constructs ~~[[are]]~~ comprise instances of meta-model constructs.

18. (Currently Amended) The method of claim 17 wherein the meta-data constructs ~~[[are]]~~ comprise constructs for meta-models of asset schemas, and wherein the metric M also depends on ~~[[the]]~~ a number of meta-model constructs.

19. (Currently Amended) The method of claim 17 wherein at least one meta-model ~~[[is]]~~ comprises a schema for relational database schemas, and wherein at least one metadata construct corresponds to a table of a relational database schema.

20. (Currently Amended) The method of claim 17 wherein at least one meta-model ~~[[is]]~~ comprises a schema for XML schemas, and wherein at least one metadata construct corresponds to a complex element of an XML schema.

21. (Currently Amended) A method, implemented in a data processing system, for determining a complexity of a metadata repository including a plurality of metadata constructs, the metadata constructs being instances of meta-model constructs for meta-models of schemas for data assets, the method comprising:

receiving a quantity of distinct meta-model constructs, denoted by C ;

evaluating a metric of complexity, denoted by M , for a metadata repository having a capacity corresponding to C , wherein the metric of complexity is evaluated according to a formula

$$M = f(C, X),$$

where f is a real-valued function of one or more real-valued parameters and X denotes ~~optional~~ zero or more additional parameters; and

using the metric M within a transaction processing system, for license of the metadata repository.

22. (Currently Amended) The method of claim 21 wherein at least one meta-model ~~[[is]]~~ comprises a schema for relational database schemas, and wherein at least one metadata construct corresponds to a table of a relational database schema.

23. (Currently Amended) The method of claim 21 wherein at least one meta-model ~~[[is]]~~ comprises a schema for XML schemas, and wherein at least one metadata construct corresponds to a complex element of an XML schema.

24. (Original) The method of claim 21 wherein the meta-model constructs include descriptors for data assets.

25. (Currently Amended) The method of claim 21 wherein the ~~optional~~ zero or more additional parameters include a parameter for a number of users of the repository.

26. (Currently Amended) The method of claim 21 wherein the ~~optional~~ zero or more additional parameters include a parameter for a number of available features.

27. (Currently Amended) The method of claim 26 wherein an available feature ~~[[is]]~~ comprises ~~[[the]]~~ an ability to change a meta-model.

28. (Currently Amended) The method of claim 26 wherein an available feature ~~[[is]]~~ comprises ~~[[the]]~~ an ability to batch scan metadata.

29. (Currently Amended) A system for determining a complexity of an enterprise information resource management system, the enterprise information resource management system being used to contain an ontology into which a plurality of enterprise data assets are

mapped, the ontology including a plurality of model constructs, the enterprise data assets including a plurality of assets constructs, and the mappings between the enterprise data assets and the ontology including a plurality of mapping constructs, the method comprising:

an input device for receiving (i) a quantity of distinct asset constructs, denoted by C_{ASSET} , (ii) a quantity of distinct mapping constructs, denoted by $C_{MAPPING}$, and (iii) a quantity of distinct model constructs, denoted by C_{MODEL} ;

a processor coupled to said input device, the processor for evaluating a metric of complexity, denoted by M , for ~~[[an]]~~ the enterprise information resource management system with capacity corresponding to C_{ASSET} , $C_{MAPPING}$, and C_{MODEL} , wherein the metric of complexity is evaluated according to a formula

$$M = f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X),$$

where f is a real-valued function of three or more real-valued parameters and X denotes ~~optional~~ zero or more additional parameters; and

a transaction processing system for receiving the metric M and for using the metric M for licensing the enterprise information resource management system.

30. (Currently Amended) The system of claim 29 wherein the plurality of enterprise data assets include assets that conform to a general data schema that uses element group asset constructs and element asset constructs.

31. (Currently Amended) The system of claim 30 wherein the general data schema ~~[[is]]~~ comprises a relational database schema, ~~[[and]]~~ the element group asset constructs ~~[[are]]~~ comprise database tables, and the element asset constructs ~~[[are]]~~ comprise columns of database tables.

32. (Currently Amended) The system of claim 30 wherein the general data schema ~~[[is]]~~ comprises an XML schema, ~~[[and]]~~ the element group asset constructs ~~[[are]]~~ comprise XML complex types, and the element asset constructs ~~[[are]]~~ comprise XML elements.

33. (Currently Amended) The system of claim 30 wherein the general data schema [[is]] comprises a Cobol Copy Book, [[and]] the element group asset constructs [[are]] comprise Cobol group items, and the element asset constructs [[are]] comprise Cobol elementary items.

34. (Currently Amended) The system of claim 30 wherein the ontology [[is]] comprises an ontology model, [[and]] wherein the model constructs include ontology classes and [[their]] properties of the ontology classes, and wherein the model constructs further include business rules that inter-relate the properties.

35. (Currently Amended) The system of claim 34 wherein the mapping constructs include mappings of element group asset constructs into ontology classes and further include mappings ~~mapping~~ of element asset constructs into ontology properties.

36. (Original) The system of claim 34 wherein the function $f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X)$, is a step function of the form

$$f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X) = f_n(X), \text{ if } C_{n-1} < C \leq C_n,$$

where C is the total number of constructs, $C = C_{ASSET} + C_{MAPPING} + C_{MODEL}$ and where C_0, C_1, C_2, \dots are cutoff points.

37. (Original) The system of claim 34 wherein the function $f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X)$ is a step function of the form

$$f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X) = f_n(X), \text{ if } C_{n-1} < C \leq C_n,$$

where C is a weighted average $C = W_{ASSET} C_{ASSET} + W_{MAPPING} C_{MAPPING} + W_{MODEL} C_{MODEL}$, and where $W_{ASSET}, W_{MAPPING}$ and W_{MODEL} are respective weighting factors, and where C_0, C_1, C_2, \dots are cutoff points.

38. (Currently Amended) The system of claim 29 wherein the enterprise information resource management system generates results for tasks, and wherein the metric M also depends on ~~[[the]]~~ a number of distinct results generated and saved.

39. (Original) The system of claim 38 wherein the results include data transformations.

40. (Original) The system of claim 38 wherein the results include SQL queries.

41. (Original) The system of claim 38 wherein the results include XSLT scripts.

42. (Currently Amended) The system of claim 29 wherein the enterprise information resource management system generates scripts for producing reports, and wherein the metric M also depends on ~~[[the]]~~ a number of distinct report scripts generated and saved.

43. (Currently Amended) The system of claim 29 wherein the enterprise information resource management system records metadata, and wherein the metric M also depends on ~~[[the]]~~ a number of distinct metadata records.

44. (Currently Amended) A system for determining a complexity of a metadata repository including a plurality of metadata constructs, the system comprising:

an input device for receiving a quantity of distinct metadata constructs, denoted by C ;
a processor coupled to said input device, the processor for evaluating a metric of complexity, denoted by M , for a metadata repository having a capacity corresponding to C , according to a formula

$$M = f(C, X),$$

where f is a real-valued function of one or more real-valued parameters and X denotes ~~optional~~ zero or more additional parameters; and

a transaction processing system for using the metric M for licensing the metadata repository.

45. (Currently Amended) The system of claim 44 wherein the plurality of metadata constructs ~~[[are]]~~ comprise instances of meta-model constructs.
46. (Currently Amended) The system of claim 45 wherein the meta-data constructs ~~[[are]]~~ comprise constructs for meta-models of asset schemas, and wherein the ~~price P~~ metric M also depends on ~~[[the]]~~ a number of meta-model constructs.
47. (Currently Amended) The system of claim 45 wherein at least one meta-model ~~[[is]]~~ comprises a schema for relational database schemas, and wherein at least one metadata construct corresponds to a table of a relational database schema.
48. (Currently Amended) The system of claim 45 wherein at least one meta-model ~~[[is]]~~ comprises a schema for XML schemas, and wherein at least one metadata construct corresponds to a complex element of an XML schema.
49. (Currently Amended) A system for determining a complexity of a metadata repository including a plurality of metadata constructs, the metadata constructs being instances of meta-model constructs for meta-models of schemas for data assets, the system comprising:
- an input device for receiving a quantity of distinct meta-model constructs, denoted by C ;
 - a processor coupled to said input device, the processor for evaluating a metric of complexity, denoted by M , for a metadata repository having a capacity corresponding to C , wherein the metric of complexity is evaluated according to a formula

$$M = f(C, X),$$

where f is a real-valued function of one or more real-valued parameters and X denotes ~~optional~~ zero or more additional parameters; and

a transaction system for receiving the metric M , the transaction system further for using the metric M for license of the metadata repository.

50. (Currently Amended) The system of claim 49 wherein at least one meta-model *[[is]]* comprises a schema for relational database schemas, and wherein at least one metadata construct corresponds to a table of a relational database schema.

51. (Currently Amended) The system of claim 49 wherein at least one meta-model *[[is]]* comprises a schema for XML schemas, and wherein at least one metadata construct corresponds to a complex element of an XML schema.

52. (Original) The system of claim 49 wherein the meta-model constructs include descriptors for data assets.

53. (Currently Amended) The system of claim 49 wherein the ~~optional~~ zero or more additional parameters include a parameter for a number of users of the repository.

54. (Currently Amended) The system of claim 49 wherein the ~~optional~~ zero or more additional parameters include a parameter for a number of available features.

55. (Currently Amended) The system of claim 54 wherein an available feature *[[is the]]* comprises an ability to change a meta-model.

56. (Currently Amended) The system of claim 54 wherein an available feature *[[is the]]* comprises an ability to batch scan metadata.

57. (Currently Amended) A computer-readable storage medium storing program code for causing a computer to determine a complexity of an enterprise information resource management system, the enterprise information resource management system being used to contain an ontology into which a plurality of enterprise data assets are mapped, the ontology including a plurality of model constructs, the enterprise data assets including a plurality of assets constructs, and the mappings between the enterprise data assets and the ontology including a plurality of mapping constructs, wherein the program code causes the computer to determine the complexity by performing the steps of:

determining (i) a quantity of distinct asset constructs, denoted by C_{ASSET} , (ii) a quantity of distinct mapping constructs, denoted by $C_{MAPPING}$, and (iii) a quantity of distinct model constructs, denoted by C_{MODEL} ;

evaluating a metric of complexity, denoted by M , for ~~[[an]]~~ the enterprise information resource management system having a capacity corresponding to C_{ASSET} , $C_{MAPPING}$, and C_{MODEL} wherein the metric of complexity is evaluated according to a formula

$$M = f(C_{ASSET}, C_{MAPPING}, C_{MODEL}, X),$$

where f is a real-valued function of three or more real-valued parameters and X denotes ~~optional~~ zero or more additional parameters; and

using the metric M within a transaction processing system, for license of the enterprise information resource management system.

58. (Currently Amended) A computer-readable storage medium storing program code for causing a computer to determine a complexity of a metadata repository including a plurality of metadata constructs, wherein the program code causes the computer to determine the complexity by performing the steps of:

determining a quantity of distinct metadata constructs, denoted by C ;

evaluating a metric of complexity, denoted by M , for a metadata repository having a capacity corresponding to C , wherein the metric of complexity is evaluated according to a formula

$$M = f(C, X),$$

where f is a real-valued function of one or more real-valued parameters and X denotes ~~optional~~ zero or more additional parameters; and

using the metric M within a transaction processing system, for license of the metadata repository.

59. (Currently Amended) A computer-readable storage medium storing program code for causing a computer to determine a complexity of a metadata repository including a plurality of metadata constructs, the metadata constructs being instances of meta-model constructs for meta-models of schemas for data assets, wherein the program code causes the computer to determine the complexity by performing the steps of:

determining a quantity of distinct meta-model constructs, denoted by C ;

evaluating a metric of complexity, denoted by M , for a metadata repository having a capacity corresponding to C , wherein the metric of complexity is evaluated according to a formula

$$M = f(C, X),$$

where f is a real-valued function of one or more real-valued parameters and X denotes ~~optional~~ zero or more additional parameters; and

using the metric M within a transaction processing system, for license of the metadata repository.

60. (Currently Amended) The method of claim 1 further comprising limiting the complexity of the enterprise information resource management system ~~so as not to exceed~~ to a specified limit by restricting the quantities C_{ASSET} , $C_{MAPPING_L}$ and C_{MODEL} .

61. (Previously Presented) The method of claim 60 wherein the specified limit is determined from a license key for the enterprise information resource management system.

62. (Previously Presented) The method of claim 60 wherein different versions of the enterprise information resource management system have different specified limits.

63. (Currently Amended) The method of claim 16 further comprising limiting the complexity of the enterprise information resource management system ~~so as not to exceed~~ to a specified limit by restricting the quantity C .

64. (Previously Presented) The method of claim 63 wherein the specified limit is determined from a license key for the enterprise information resource management system.
65. (Previously Presented) The method of claim 63 wherein different versions of the enterprise information resource management system have different specified limits.
66. (Currently Amended) The method of claim 21 further comprising limiting the complexity of the enterprise information resource management system ~~so as not to exceed~~ to a specified limit, by restricting the quantity C .
67. (Previously Presented) The method of claim 66 wherein the specified limit is determined from a license key for the enterprise information resource management system.
68. (Previously Presented) The method of claim 66 wherein different versions of the enterprise information resource management system have different specified limits.
69. (Currently Amended) The system of claim 29 further comprising a controller restricting the quantities C_{ASSET} , $C_{MAPPING_L}$ and $C_{MODEL}[[,]]$ ~~as to ensure~~ such that the complexity of the enterprise information resource management system does not exceed a specified limit.
70. (Previously Presented) The system of claim 69 wherein the specified limit is determined from a license key for the enterprise information resource management system.
71. (Previously Presented) The system of claim 69 wherein different versions of the enterprise information resource management system have different specified limits.
72. (Currently Amended) The system of claim 44 further comprising a controller restricting the quantities $C[[,]]$ ~~so as to ensure~~ such that the complexity of the enterprise information resource management system does not exceed a specified limit.

73. (Previously Presented) The system of claim 72 wherein the specified limit is determined from a license key for the enterprise information resource management system.

74. (Previously Presented) The system of claim 72 wherein different versions of the enterprise information resource management system have different specified limits.

75. (Currently Amended) The system of claim 49 further comprising a controller restricting the quantities $C[[,]]$ ~~so as to ensure~~ such that the complexity of the enterprise information resource management system does not exceed a specified limit.

76. (Previously Presented) The system of claim 75 wherein the specified limit is determined from a license key for the enterprise information resource management system.

77. (Previously Presented) The system of claim 75 wherein different versions of the enterprise information resource management system have different specified limits.